ISSN: 2586-7652 (Print) Vol. 04, No. 01, March 2021
ISSN: 2635-7607 (Online)
International Journal of Advanced Engineering
Source: http://ictaes.org
Manuscript received: August 5, 2020; Revised: February 5, 2021; Accepted: February 8, 2021

Monitoring Human Body in Video surveillance

Bikram Shah[1], Sabin Kafley[2], Manoj K. Guragai[3]

[1,2,3]*Department of Electronics and Computer Engineering, Purwanchal Campus,Institute of Engineering, Tribhuwan University, Dharan, Nepal*

bikramshah213@gmail.com[1], sabinkafley@ioepc.edu.np[2] ,mkguragai@gmail.com[3]

Abstract

Video monitoring are used to track critical parts of defense. Such as banks, restaurants, crowded areas and boundaries. Advanced computer power, ubiquitous storage equipment and high-speed network infrastructure paved the way for more cheap multi-sensor video surveillance systems. The video production is historically processed online by human operators and normally only after a forensic incident is saved to tapes. To support human operators in identifying critical events in video using an intelligent video surveillance device, filtering out redundant information generated by a number of cameras and increasing the response times to forensic events has become an essential necessity. The development of intelligent video monitoring systems needs fast, reliable and robust algorithms to transfer human detection, classification, tracking, and activity analysis. The paper introduces an intelligent visual control device with moving human identification, classification and tracking capabilities in real time. Indoor and outdoor settings and under changing lighting conditions it can handle the identification of people.

Keywords: *video-surveillances; identification; classification; tracking, sum of absolute difference (sad)*

1. Introduction

Video surveillance is an active computer-vision research topic aimed at identifying, recognizing and monitoring images via a sequence. It attempts to recognize and explain object actions by substituting human operators for the old conventional way of tracking cameras. The identification of objects requires identifying objects within the video sequence frame. Object tracking is an object or multiple object position process with a camera over time. The main steps in video analysis are the identification of interesting moving objects, the following of such objects from each frame to frame, and the analysis of object tracks to identify their actions [1].

The use of object tracking is important for the detection of motion. A variable number of objects is important

Corresponding Author : Bikram Shah
Author's affiliation: Dept. of Electronics and Computer Engineering, Purwanchal Campus, IoE, TU, Nepal
Email: bikramshah213@gmail.com
Copyright © ICT-AES

for automated detection, monitoring and counting. In monitoring algorithms, there are some difficult problems, including context modelling, extraction functionality, tracking, occlusion handling and event detection. In addition, algorithms for computer vision are still not stable enough to manage fully automatic systems and research is still being carried out into such improvements. This work focuses on developing a system for detecting moving objects and creating accurate video tracks.

Various background subtraction methods are available for moving object detection. The absolute discrepancy between the current picture and the reference modified background over a period of time is involved in background subtraction. A successful background subtraction should be able to solve the problem of various lighting conditions, background clutter, shadows, camouflage, and bootstrapping and real-time motion segmentation of the foreground object at the same time. In one history subtraction technique, it is difficult to get all these issues solved. So the concept was to model and measure their success in dynamic scenarios on different video data taken. In the presence of uncertainty, object tracking is a very difficult task: lighting state, background motion, complex object form, partial and complete object occlusions. Modification is done here in this paper to overcome the issue of variance in lighting and background clutter such as artificial motion due to the leaves of the trees, water flowing, or waving flag in the wind.

2. The Methodology

Object tracking is the method of finding a moving object in a video frame series and tracking it. To record the video, smart cameras are used as input sensors. There could be some disturbance in the captured video due to bad weather (light, wind, etc. or due to problems in sensors). Few algorithms are tested to improve the quality of the image, to detect moving objects, to measure the distance and speed of moving objects. The extraction of objects using characteristics is referred to as object detection. Based on its dimensions, every object has a particular function. Applying the feature extraction algorithm, we can point out the object in each frame.

The method of locating a moving object in a frame series is known as tracking. This monitoring can be done by using the object extraction function and by detecting artifacts in frame sequences. We can measure the moving object's location and velocity by using the object's position values in each frame.

The main objective of tracking and detecting objects (players) is to create in consecutive frames a correspondence between object parts and to extract information about objects such as trajectory, posture, speed and direction. It is a significant and difficult task to monitor objects identified frame by frame in video. It is a vital component of smart surveillance systems because it would not be possible to extract coherent temporal information about objects without object detection, and higher level behavior analysis measures would not be possible. The first step in video analysis is moving object detection.

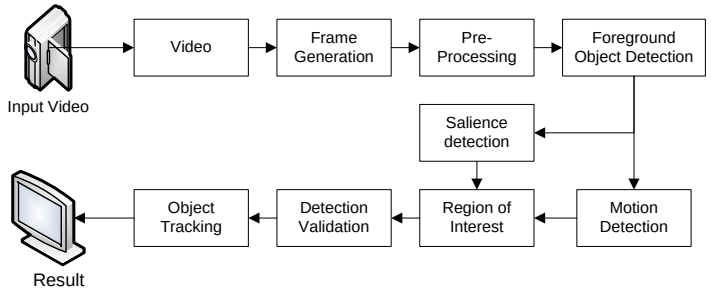


Figure 1. Architecture of proposed system

2.1 Pre-processing

Human perceive colour through RGB cone. By emitting a controlled combination of these three basic colours (red, green and blue), and hence stimulate the three types of cones at will, we are able to generate almost any perceivable colour. This is the reasoning behind why colour images are often stored as three separate image matrices; one storing the amount of red (R) in each pixel, one the amount of green (G) and one the amount of blue (B). We call such colour images as stored in an RGB format.

In grey scale images, however, we do not differentiate how much we emit of the different colours; we emit the same amount in each channel. What we can differentiate is the total amount of emitted light for each pixel; little light gives dark pixels and much light is perceived as bright pixels.

When converting an RGB image to gray scale, we have to take the RGB values for each pixel and make as output a single value reflecting the brightness of that pixel. One such approach is to take the average of the contribution from each channel: (R+B+C)/3. However, since the perceived brightness is often dominated by the green component, a different, more "human-oriented", method is to take a weighted average, e.g.: 0.3R + 0.59G + 0.11B. As RGB image consumes much time and many algorithms need the image to be in two dimensional, RGB image is converted into grey scale and sometimes even to binary form.

2.2 Frame Difference Method

Frame differencing is a pixel-wise differencing between two or three consecutive frames in an image sequence to detect regions corresponding to moving object such as human and vehicles. The threshold function determines change and it depends on the speed of object motion. It's hard to maintain the quality of segmentation, if the speed of the object changes significantly. Frame differencing is very adaptive to dynamic environments, but very often holes are developed inside moving entities [2].

Videos are actually consists of sequences of images, each of which called as a frame. For detecting moving objects in video surveillance system, use of frame difference technique from the difference between the current frame and a reference frame called as 'background image' is shown. That method is known as frame difference method.

Frame differencing is the simplest moving object detection method which is based on determining the difference between input frame intensities and background model by using pixel per pixel subtraction. Grad. Sch. of Eng. et al. have proposed frame difference method to detect the moving objects. In this case, frame difference method is performed on the three successive frames, which are between frame Fk and Fk−1 and also the frame between Fk & Fk+1 and the output image as frame difference image is two difference images dk−1 and dk+1 is expressed as

$$d_{k-1} = |f_k - f_{k-1}| \quad (1)$$

$$d_{k+1} = |f_k - f_{k+1}| \quad (2)$$

$$d_k'(x; y) = 1; \text{ if } d_k'(x; y) > T \quad (3)$$

$$0; \text{ otherwise}$$

where $k^{'} = k - 1$ *and* $k + 1$.

The process is followed by applying an operator to dk−1 and dk+1. Here original frames are shown and after preprocessing segmented results frames are also shown.



Figure 2. Original video frames

2.3 Foreground Object Detection

Detection and segmentation of foreground objects from a video which contains both stationary and moving background objects and undergoes both gradual and sudden "once-off" changes. A Bayes decision rule for classification of background and foreground from selected feature vectors is formulated. Under this rule, different types of background objects will be classified from foreground objects by choosing a proper feature vector. The stationary background object is described by the color feature, and the moving background object is represented by the color co-occurrence feature. Foreground objects are extracted by fusing the classification results from both stationary and moving pixels. Learning strategies for the gradual and sudden "once-off" background changes are proposed to adapt to various changes in background through the video. The convergence of the learning process is proved and a formula to select a proper learning rate is also derived. Experiments have shown promising results in extracting foreground objects from many complex backgrounds including wavering tree branches, flickering screens and water surfaces, moving escalators, opening and closing doors, switching lights and shadows of moving objects.

2.4 Simple Background Subtraction

In simple background subtraction an absolute difference is taken between every current image $I_t(x; y)$ and the reference background image $B(x; y)$ to find out the motion detection mask $D(x; y)$ [3]. The reference background image is generally the first frame of a video, without containing foreground object.

$$D(x; y) = 1 \; ; if |I_t(x; y) - B(x; y)| \geq t \qquad (4)$$

$$0, Otherwise$$

where t is a threshold, which decides whether the pixel is foreground or background. If the absolute difference is greater than or equal to the pixel is classified as foreground, otherwise the pixel is classified as background [4].

2.5 Blob Detection

Blob detection methods are aimed at detecting regions in a digital image that differ in properties, such as

brightness or color, compared to surrounding regions. Informally, a blob is a region of an image in which some properties are constant or approximately constant; all the points in a blob can be considered in some sense to be similar to each other. Given some property of interest expressed as a function of position on the image, there are two main classes of blob detectors: (i) differential methods, which are based on derivatives of the function with respect to position, and (ii) methods based on local extrema, which are based on finding the local maxima and minima of the function. With the more recent terminology used in the field, these detectors can also be referred to as interest point operators, or alternatively interest region operators.

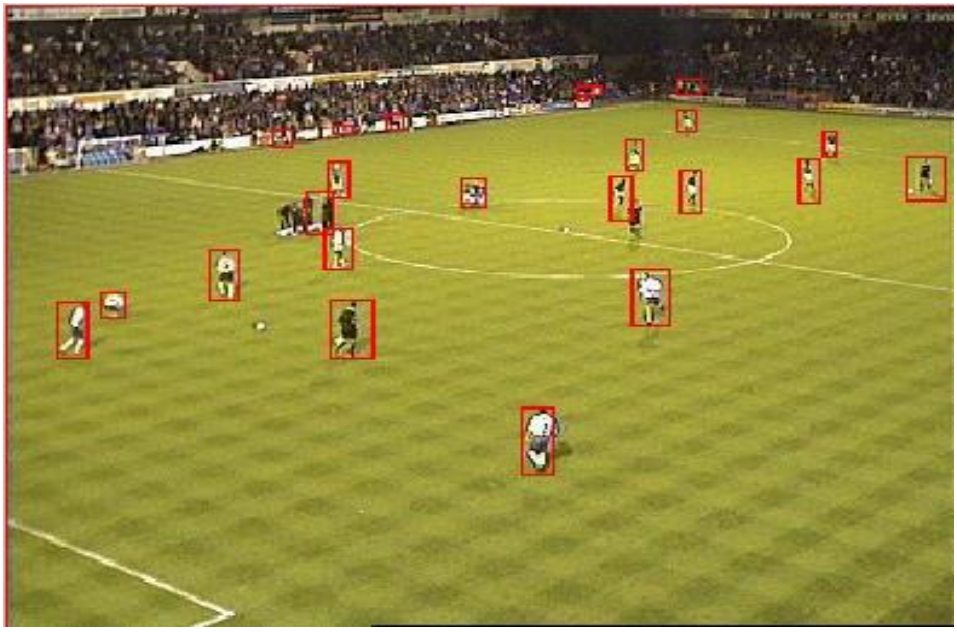


Figure 3. Blob Detection (source: www.espnsports.com)

There are several motivations for studying and developing blob detectors. One main reason is to provide complementary information about regions, which is not obtained from edge detectors or corner detectors. In early work in the area, blob detection was used to obtain regions of interest for further processing. These regions could signal the presence of objects or parts of objects in the image domain with application to object recognition and/or object tracking. In other domains, such as histogram analysis, blob descriptors can also be used for peak detection with application to segmentation. Another common use of blob descriptors is as main primitives for texture analysis and texture recognition. In more recent work, blob descriptors have found increasingly popular use as interest points for wide baseline stereo matching and to signal the presence of informative image features for appearance-based object recognition based on local image statistics. There is also the related notion of ridge detection to signal the presence of elongated objects.

One of the first and also most common blob detectors is based on the Laplacian of the Gaussian (LoG). Given an input image, $f(x,y)$ which is convolved by a Gaussian kernel using Equation (5) at a certain scale t to give a scale space representation $L(x, y, t)= g(x, y, t)*f(x, y)$.

$$g\,(x, y, t)= \frac{1}{2\pi t^2} e^{\frac{x^2+y^2}{-2t^2}} \tag{5}$$

Then, the result of applying the Laplacian operator $\nabla^2 L = L_{xx} + L_{yy}$ is computed, which usually results in strong positive responses for dark blobs of extent $\sqrt{2t}$ and strong negative responses for bright blobs of similar

size. A main problem when applying this operator at a single scale, however, is that the operator response is strongly dependent on the relationship between the size of the blob structures in the image domain and the size of the Gaussian kernel used for pre-smoothing. In order to automatically capture blobs of different (unknown) size in the image domain, a multi-scale approach is therefore necessary. A straightforward way to obtain a multi-scale blob detector with automatic scale selection is to consider the scale-normalized Laplacian operator.

$$\nabla_{\text{norm}}{}^2 \text{L} = \text{t}(\text{L}_{xx} + \text{L}_{yy}) \tag{6}$$

And to detect scale-space maxima/minima, that are points that are simultaneously local maxima/minima of $\nabla_{\text{norm}}{}^2 \text{L}$ with respect to both space and scale. Thus, given a discrete two-dimensional input image f(x,y) a three-dimensional discrete scale-space volume L(x, y, t) is computed and a point is regarded as a bright (dark) blob if the value at this point is greater (smaller) than the value in all its 26 neighbors. Thus, simultaneous selection of interest points $(\hat{x}, \hat{y})$ and scales $\hat{t}$ is performed according to

$$(\hat{x}, \hat{y}, \hat{t}) = \text{argmaxminlocal}_{(x,y,t)} (\nabla_{\text{norm}}{}^2 \text{L}) (x, y, t). \tag{7}$$

Note that this notion of blob provides a concise and mathematically precise operational definition of the notion of "blob", which directly leads to an efficient and robust algorithm for blob detection. Some basic properties of blobs defined from scale-space maxima of the normalized Laplacian operator are that the responses are covariant with translations, rotations and rescalings in the image domain. Thus, if a scale-space maximum is assumed at a point(x_0,y_0;t_0) then under a rescaling of the image by a scale factor s, there will be a scale-space maximum at (sx_0,sy_0,s^2t_0) in the rescaled image . This in practice highly useful property implies that besides the specific topic of Laplacian blob detection, local maxima/minima of the scale-normalized Laplacian are also used for scale selection in other contexts, such as in corner detection, scale-adaptive feature tracking, in the scale-invariant feature transform (Lowe 2004) as well as other image descriptors for image matching and object recognition.



Figure 4. Object detection using blob (Source: 2007.ICICIC'07) PP. 408-408, 2007)

2.6 Motion Detection

An automatic video surveillance is used by private companies, governments and public organizations to fight against terrorism and crime, public safety in airports, bus stand, railway station, town centers and hospitals. It has also found applications in traffic surveillance for efficient management of transport networks and road safety. Video surveillance system include task such as motion detection, tracking, and activity recognition. Out of the task

mentioned above, detection of moving object is the first important step and successful segmentation of moving foreground object from the background ensures object classification, personal identification, tracking, and activity analysis, making these later step more efficient. Hu et al. categorized motion detection into three major classes of method as frame differencing, background subtraction and Gaussian mixture.

2.7 Object tracking

Object tracking is the process of locating an object or multiple objects over time using a camera [5]. Object tracking, in general, is a challenging problem. Difficulties in tracking objects can arise due to abrupt object motion, changing appearance patterns of the object and the scene, non-rigid object structures, object-to-object and object-to-scene occlusions, and camera motion. Tracking is usually performed in the context of higher-level applications that require the location and/or shape of the object in every frame [6].

Typically, assumptions are made to constrain the tracking problem in the context of a particular application. In this survey, we categorize the tracking methods on the basis of the object and motion representations used, provide detailed descriptions of representative methods in each category, and examine their pros and cons. Moreover, we discuss the important issues related to tracking including the use of appropriate image features, selection of motion models, and detection of object.

2.7.1 The Sum of Absolute Difference Algorithm

The SAD algorithm is widely used method for motion estimation, which finds widespread application in areas like image registration for Automated Video Surveillance and video compression using H.264, it forms the most computationally intensive part of image registration as well as video compression. In the process of video coding, the similarities between video frames could be used to achieve higher compression ratios. The usual coding techniques applied to moving objects within a video scene lower the compression efficiency as they only consider the pixels at the same position in the video frames. Motion estimation and the SAD algorithm are used to capture such movements more accurately for better compression efficiency. In video surveillance using moving cameras, a popular way to handle translation problems on images, using template matching is to compare the intensities of the pixels, using the SAD measure [7].

The motion estimation on a video sequence using SAD uses the current video frame and a previous frame as the target frame. The two frames are compared pixel by pixel, summing up the absolute values of the differences of each of the two corresponding pixels. The result is a positive number that is used as the score. SAD reacts very sensitively to even minor changes within a scene.

The block based motion estimation is performed on a set of pixels, every frame is divided into equally sized blocks, and for each block in the current frame, a search for most resembling block is done in the reference frame, searching the whole reference frame for each block in the current frame makes this task computationally intensive. After the best matches are found for the current block the motion vectors are stored along with the SAD values.

The basic unit of this method is a macro block of size 16×16 or 8×8 on the previous frame (reference frame), which is taken as the reference to gauge the direction of motion. SAD algorithm is usually implemented in a nested loop with conditional branch.

```
for (rows of macro blocks)
        {
```

```
for (Columns of macro blocks)
        {
                for(for(row of template)
                {
                 for(columns of template)
                        {
                        SAD Computation;
                        SAD Comparison ;(> threshold)
                        }
                }
        }
}
```

The Besides the calculation of SAD there are four loops in the algorithm. It seeks the best matched macro block in the search region of the reference image frame to the current macro block in the current image frame. The best matched macro block is the one which has the minimum SAD value with the current macro block. The algorithm depicted is computationally intensive as very large number of calculations and comparisons are required. However, as it is a block-based algorithm, it is a good candidate for being parallelized and implemented on parallel architectures like the GPU architecture.

2.7.2 Our approach to SAD

The In our approach to the SAD, the image is divided into macro blocks, with the SAD computation for each macro block being handled by a thread. Thus for an image size of 1024×768 the thread configuration for a macro block size of 8×8 was 128×96 threads, and 64×48 for a 16×16 macro block, with each thread performing the computation for its respective macro block. In the reference frame a 32×32 (or 16×16) region called the window, is selected and the sum of absolute differences (SAD) is computed in this window frame, with the macro block of the current frame i.e. one by one all the possible 16×16 (or 8×8) blocks on the 32×32 (or 16×16) window frame are taken for SAD computation on each pixel. The frame in which the SAD turns out to be the minimum is selected and determines the motion of the object from the reference frame to the current frame.

In the following section we describe the approach taken for the image size of 256×191. For this image size we have used 8×8 as the macro block size so shared memory could be used effectively, and portions of both the frames that were required for the SAD computation could be accessed from the shared memory which is limited to 6kilobytes per block. As the computation of SAD on each pixel refrains from accessing memory locations within the same banks, there were lesser bank conflicts. As the shared memory is limited, the portions of the current and the reference frames required for SAD computation of the macro blocks could not be accommodated in the shared memory, to overcome this.

2.7.3 Shape based Human Detection and Tracking

We learn full-body detectors for walking or standing humans by the method proposed. Nested structure detectors are learned by boosting edge let feature based weak classifiers. To cover different viewpoints, two detectors are learned: one for the left profile view, and one for the frontal/rear view (the detector for right profile view is generated by flipping the left profile view horizontally). The training set contains 1,700 positive samples for frontal/rear views, 1,120 for left profile view, and 500 negative images. The negative images are all of street scenes. The samples in training set are independent of the test sequences. We do not use the combined detection in for partial occlusion reasoning explicitly, as the local feature based full-body detector can work with partial occlusion to some extent and inter-human occlusions are not strong in this data set.

We constrain the search of humans around moving blobs. Motion is detected by comparing pixel colour to an adaptively learned background model. If the proportion of the moving pixels within an image sub-window is larger than a threshold, μm, it is considered a candidate for human hypotheses and sent to the detector for further process; otherwise it is discarded directly. This reduces the search space of the human detector and prevents false alarms on static scene objects; however, it also prevents detection of static persons in the scene (in aeri l surveillance scenario, persons will always be expected to enter the scene at some time). Humans are tracked by forming associations between the frame detection responses. This 2D tracking method is a simplified version, as only the full-body detector is applied. The affinity between a hypothesis and a response is calculated based on cues from distance, size, and colour. A greedy algorithm is used to associate the hypotheses and the detection responses. The automatic initialization and termination of trajectories are based on the confidences calculated from associated detection responses. To track the human, first data association with the full-body detection responses is attempted; if this fails, a colour based mean shift tracker is used to follow the person.

3. Result and Discussion

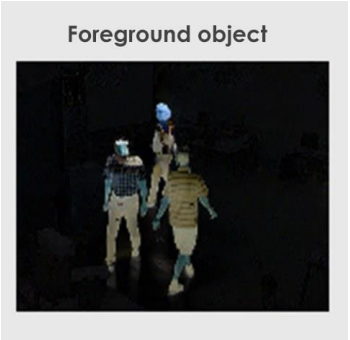
This research focuses on the identification and monitoring of video surveillance of the human body. For instance, the frames are initially created from the video as shown in Figure 4. The background and foreground are divided to find the area of interest using our model. The region of interest here means the area containing only the human body in the video. Human validation is conducted and monitored in video surveillance following the identification of the human body. The observations are shown in the following Figures.



(a)



(b)



(c)



(d) (e) (f)

Figure 4. (a) Video Frames, (b) Background image (c), Foreground object, (d) Region of interest, (e) Human validation (f) Detected human region

4. Conclusion

We have introduced a new tracking-by-detection structure for online object tracking in this paper. We start with the development of a series of proposals to locate artifacts extracted from a learning detector model. The contributions of this research is it can analyze many meaningful human's motion and event, also can distinguish single human and multiple human. Also it can track people through occlusions. Thus, we have successfully implemented our project and also hope that our system proposed would be better than that already in place as our system proposed has a 90.51 percent reliability and the potential to solve the drawbacks of the current system.

References

[1] Weiming, H., Tieniu, T., Liang, W., and Steve, M. (2004). A survey on visual surveillance of object motion and behaviors. Systems, Man, and Cybernetics, Part C*:* Applications and Reviews, IEEE Transactions on, 34(3):334–352

[2] Zhan, C., Duan, X., Xu S., Song, Z., and Luo, M. An improved moving object detection algorithm based on frame difference and edge detection (2007). In Image and Graphics, 2007. ICIG 2007. Fourth International Conference on, pages 519–523. IEEE

[3] Cezar, J., Silveira J., Claudio, R. J., and Soraia R. M. (2005.) Background subtraction and shadow detection in grayscale video sequences. In *C*omputer Graphics and Image Processing, 2005. SIBGRAPI 2005. 18th Brazilian Symposium on, pages 189–196. IEEE

[4] Alper, Y., Omar J., and Mubarak, S. Object tracking: A survey (2006). Acm Computing Surveys (CSUR), 38(4):13

[5] Budi, S., Hyoungseop, K., Joo, K. T., and Seiji, I.. Tracking of moving objects by using a low resolution image. In Innovative Computing, Information and Control, (2007). ICICIC'07 (IEEE), 2nd International Conference on, PP. 408–408.

[6] Alan, J. L. , Hironobu, F. and Raju, S. P. , Moving Target Classification and tracking from real time videos.

[7] Ya, L., Haizhou, A., and Guang-you, Xu. Moving object detection and tracking based on background subtraction (2001). In Multispectral Image Processing and Pattern Recognition, pages 62–66. International Society for Optics and Photonics.